

Listing of Claims:

1. (currently amended) A computer implemented method for determining positional relationships among objects represented in a database, the method comprising:

receiving a query including at least a first object in the database and a desired positional relationship between the first object and a second object in the database;

in response to receiving the query performing the steps of:

identifying, for the first object, one or more tiles in a plurality of tiles defined in a spatial index in the database that ~~intersect and are within the boundaries~~ are interior tiles of the first object and one or more tiles in the plurality of tiles that ~~intersect a boundary~~ are boundary tiles of the first object;

identifying, for the second object, one or more tiles in the plurality of tiles that ~~intersect and are within the boundaries~~ are interior tiles of the second object and tiles in the plurality of tiles that intersect ~~intersect a boundary~~ are boundary tiles of the second object; and

providing the query to a primary filter operable to:

compare tiles of the first object with tiles of the second object to determine whether at least one tile of the first object intersects with at least one tile of the second object; and

determine whether the at least one tile of the second object with which the at least one tile of the first object intersects is a an interior tile ~~that intersects the second object while all of the boundary of the second object lies outside the tile;~~

if so, providing the first object to ~~the~~ a results set of objects that satisfy the desired positional relationship,

otherwise, providing the first object and the second object to a secondary filter operable to:

compare the second object with the first object; and

if the first object fulfills the desired positional relationship, include the first object in the result set of objects that satisfy the desired positional relationship;

otherwise, exclude the first object from the result set of objects that satisfy the desired positional relationship.

2. (canceled)

3. (canceled)

4. (currently amended) The method according to claim 1, wherein ~~[[a]]~~ the desired positional relationship among objects includes at least one member selected from the group comprising:

[[a]] the first object in [[a]] the second object,

[[a]] the second object contained by [[a]] the first object,

[[a]] the first object in [[a]] the second object where a boundary of the first object intersects a boundary of the second object,

[[a]] the first object covering [[a]] the second object where a boundary of the first object intersects a boundary of the second object,

[[a]] the first object outside [[a]] the second object where a boundary of the first object intersects a boundary of the second object,

[[a]] the first object does not cover [[a]] the second object where a boundary of the first object intersects a boundary of the second object,

[[a]] the first object overlaps [[a]] the second object but the boundaries of the first object and the second object do not touch,

[[a]] the first object overlaps [[a]] the second object and the boundaries of the first object and the second object touch,

[[a]] the first object is equivalent to [[a]] the second object, and

[[a]] the first object is disjoint from [[a]] the second object.

5. (canceled)

6. (canceled)

7. (original) The method according to claim 1, wherein the tiles all have at least one of the same size and the same shape.

8. (original) The method according to claim 7, wherein the tiles are rectangular.

9. (original) The method according to claim 1, wherein the tiles have different sizes.

10. (previously presented) The method according to claim 1, wherein the database comprises objects on a surface or in a three-dimensional space.

11. (previously presented) The method according to claim 10, wherein the database stores exact representations of the objects and approximate representations of the objects.

12. (previously presented) The method according to claim 1, wherein the objects comprise polygons and the method determines positional relationships among polygons.

13. (previously presented) The method according to claim 11, wherein the objects comprise linestrings and polygons and the method determines positional relationships among the polygons.

14. (original) The method according to claim 13, wherein the linestrings represent streets and the polygons represent a geographic region, and the method determines streets that at least partially fall within the geographic region.

15. (original) The method according to claim 14, wherein the geographic region includes a member selected from the group comprising counties, cities, and zip code regions.

16. (canceled)

17. (previously presented) The method according to claim 1, wherein the objects are organized with a linear quadtree index.

18. (previously presented) The method according to claim 1, wherein the step of determining whether the at least one tile of the second object with which the at least one tile of the first object intersects is an interior tile of the second

object is applied only to objects wherein at least one tile of the first object intersects with at least one tile of the a second object.

19. (previously presented) The method according to claim 1, further comprising:

assigning each tile, a tile code and an associated object as the tiles are defined.

20. (currently amended) A computer program product for performing a process for determining positional relationships among objects represented in a database in a computer system, comprising:

a computer readable medium; and
computer program instructions, recorded on the computer readable medium, executable by a processor, for performing the steps of
receiving a query including at least a first object in the database and a desired relationship between the first object and a second object in the database;
in response to receiving the query performing the steps of:

identifying, for the first object, one or more tiles in a plurality of tiles defined in a spatial index in the database that ~~intersect and are within the boundaries~~ are interior tiles of the first object and one or more tiles in the plurality of tiles that ~~intersect a boundary~~ are boundary tiles of the first object

identifying, for the second object, one or more tiles in the plurality of tiles that ~~intersect and are within the boundaries~~ are interior tiles of the second object and tiles in the plurality of tiles that intersect ~~intersect a boundary~~ are boundary tiles of the second object; and

providing the query to a primary filter operable to:

compare tiles of the first object with tiles of the second object to determine whether at least one tile of the first object intersects with at least one tile of the second object; and

determine whether the at least one tile of the second object with which the at least one tile of the first object intersects is a an interior tile ~~that intersects the second object while all of the boundary of the second object lies outside the tile;~~

if so, providing the first object to ~~the~~ a results set of objects that satisfy the desired positional relationship,

otherwise, providing the first object and the second object to a secondary filter operable to:

compare the second object with the first object; and

if the first object fulfills the desired positional relationship, include the first object in the result set of objects that satisfy the desired positional relationship;

otherwise, exclude the first object from the result set of objects that satisfy the desired positional relationship.

21. (currently amended) A system for performing a process method for determining positional relationships among objects represented in a database, comprising:

a processor operable to execute computer program instructions; and

a memory operable to store computer program instructions executable by the processor, for performing the steps of:

receiving a query including at least a first object in the database and a desired positional relationship between the first object and a second object in the database;

in response to receiving the query performing the steps of:

identifying, for the first object, one or more tiles in a plurality of tiles defined in a spatial index in the database that ~~intersect and are within the boundaries~~ are interior tiles of the first object and one or more tiles in the plurality of tiles that ~~intersect a boundary~~ are boundary tiles of the first object;

identifying, for the second object, one or more tiles in the plurality of tiles that ~~intersect and are within the boundaries~~ are interior tiles of the second object and tiles in the plurality of tiles that intersect ~~intersect a boundary~~ are boundary tiles of the second object; and

providing the query to a primary filter operable to:

compare tiles of the first object with tiles of the second object to determine whether at least one tile of the first object intersects with at least one tile of the second object; and

determine whether the at least one tile of the second object with which the at least one tile of the first object intersects is a an interior tile ~~that intersects the second object while all of the boundary of the second object lies outside the tile;~~

if so, providing the first object to ~~the~~ a results set of objects that satisfy the desired positional relationship,

otherwise, providing the first object and the second object to a secondary filter operable to:

compare the second object with the first object; and

if the first object fulfills the desired positional relationship, include the first object in the result set of objects that satisfy the desired positional relationship;

otherwise, exclude the first object from the result set of objects that satisfy the desired positional relationship.

22. (previously presented) The method of claim 1, further comprising the steps of:

exactly comparing the first object the second object if none of the tiles of the second object that intersect with the at least one tile of the first object is a tile

that intersects the second object while all of the boundary of the second object lies outside the tile.

23. (previously presented) The computer program product of claim 20, further comprising the steps of:

exactly comparing a geometry of the first object with the second object if none of the tiles of the second object that intersect with the at least one tile of the first object is a tile that intersects the second object while all of the boundary of the second object lies outside the tile.

24. (previously presented) The system of claim 21, further comprising the steps of:

exactly comparing the first object with the second object if none of the tiles of the second object that intersect with the at least one tile of the first object is a tile that intersects the second object while all of the boundary of the second object lies outside the tile.